10

Claims

1. A method of configuring parameters of an M2M (machine-to-machine) module, the method comprising establishing a connection between the M2M module and a server, c h a r a c t e r i z e d by the method compris-
5   ing:

(304) downloading, to the M2M module, an application having an interface for configuring the M2M module, the application being configured to run on a Java virtual machine (JVM);

(306, 308) communicating with the server by the application for re-
10   ceiving configuration parameters; and

(310) setting the parameters of the M2M module by the application based on the received configuration parameters.

2. The method of claim 1, c h a r a c t e r i z e d in that the connection between the M2M module and the server is established through a M2M
15   gateway or over a TCP/IP connection.

3. The method of claim 1, c h a r a c t e r i z e d in that the communicating with the server comprises: requesting (306) configuration parameters from the server and receiving (308), in the application, the configuration parameters from the server.

20   4. The method of claim 1, c h a r a c t e r i z e d by the application being a Java 2 Micro Edition (J2ME™) application.

5. The method of claim 3, c h a r a c t e r i z e d by the application being a Java MIDlet or a Java IMlet.

6. The method of claim 1, c h a r a c t e r i z e d by the application
25   being downloaded to the M2M module over a cable, over an infrared connection or over-the-air (OTA).

7. The method of claim 1, c h a r a c t e r i z e d by the application programming interface being a Common Object Request Broker Architecture (CORBA) API.

30   8. The method of claim 6, c h a r a c t e r i z e d by communicating with the server by making a method call through the CORBA API.

9. The method of claim 1, c h a r a c t e r i z e d by using the M2M module (110) for configuring parameters of a remote device based on the configuration parameters received from the server.

10. An M2M (machine-to-machine) module (110), comprising: means for operating a Java virtual machine (122) and means (124) for establishing a connection between the M2M module (110) and a server (118),

c h a r a c t e r i z e d   by the M2M module (110) being configured to:

download an application (120) having an application programming interface (API) (124) for configuring the M2M module, the application (120) being configured to run on a Java virtual machine (JVM) (122);

communicate with the server (118) by the application (120) for receiving configuration parameters; and

set the parameters of the M2M module (110) by the application (120) based on the received configuration parameters.

11. The M2M module of claim 10, c h a r a c t e r i z e d   in that the M2M module (110) is further configured to request, by the application, configuration parameters from the server (118) and receiving, in the application, the configuration parameters from the server (118).

12. The M2M module of claim 10, c h a r a c t e r i z e d   in that the application (120) is a Java 2 Micro Edition (J2ME™) application.

13. The M2M module of claim 10, c h a r a c t e r i z e d   in that the application (120) is a Java MIDlet or a Java IMlet.

14. The M2M module of claim 10, c h a r a c t e r i z e d   in that the M2M module (110) is configured to download the application (120) over a cable, over an infrared connection or over-the-air (OTA).

15. The M2M module of claim 10, c h a r a c t e r i z e d   in that the application-programming interface (124) is a Common Object Request Broker Architecture (CORBA) API.

16. The M2M module of claim 15, c h a r a c t e r i z e d   in that the M2M module (110) is configured to communicate with the server (118) by making a method call through the CORBA API (124).

17. The M2M module of claim 10, c h a r a c t e r i z e d   in that the M2M module (110) is used for configuring parameters of a remote device (112) based on the configuration parameters received from the server (118).